Decoy Bandits Dueling on a Poset

Julien Audiffren, Ralaivola Liva

Decoy Bandits Dueling on a Poset

Julien Audiffren
CMLA, ENS Cachan, CNRS
Paris Saclay University
94235 Cachan France
audiffren@cmla.ens-cachan.fr

Liva Ralaivola
QARMA, LIF, CNRS
Aix Marseille University
F-13289 Marseille cedex 9, France
liva.ralaivola@lif.univ-mrs.fr

Abstract

We adress the problem of dueling bandits defined on partially ordered sets, or posets. In this setting, arms may not be comparable, and there may be several (incomparable) optimal arms. We propose an algorithm, UnchainedBandits, that efficiently finds the set of optimal arms of any poset even when pairs of comparable arms cannot be *distinguished* from pairs of incomparable arms, with a set of minimal assumptions. This algorithm relies on the concept of decoys, which stems from social psychology. For the easier case where the incomparability information may be accessible, we propose a second algorithm, SlicingBandits, which takes advantage of this information and achieves a very significant gain of performance compared to UnchainedBandits. We provide theoretical guarantees and experimental evaluation for both algorithms.

1 Introduction

Chasing the optimal set for cold-start recommendation. Today's recommendation systems heavily rely on machine learning. Dedicated techniques may indeed be designed to extract statistical regularities from a set of collected behaviors and provide users with relevant recommendations. One of the main challenges a recommendation system has to deal with is *cold-start*, i.e. the situation where recommendations must be computed for a user for whom no information has been collected. A common strategy to get around this problem is to have at hand a set of default items to recommend to any new customer. The design of such a set is then paramount to the user experience with the recommendation system and to his willingness to rely on it for future movie suggestions. A natural goal is therefore to try to form a set of *best* movies. Identifying the best movies is a task that requires a proper handling of two features: a) the variety of existing film genres (documentary, drama, comedy...) and b) the uncertainty with which one film may be considered better than another. The variety of genres induces the issue of incomparability: there are pairs of movies —comparison of pairs is evidently at the core of the best movie selection process— that *cannot* be compared such as, e.g., a documentary and a horror movie. This means that movies are only *partially ordered* and it suggests that the *set* of best movies *must* contain incomparable movies. Said otherwise, each movie from the set is the best in its category. The uncertainty issue mentioned above then arises within a single genre as it might be complex to assert that a film is better than another. A way to bypass this difficulty is to rely on a committee of critics and to aggregate the (noisy) opinions of its members on pairs of comparable movies. This might be implemented as follows: for each pair of films, committee members are chosen at random and asked which of the two movies is the better, and the movie that wins the most among the random probes is decided to be the best. This introduction provides a practical motivation for the present paper where we study the question of deriving strategies for *dueling bandits* defined on *partially ordered sets*, or *posets*. We are in particular interested in being able to find the set of best arms among all the (possibly incomparable) arms at hand.

Dueling Bandits on Posets. Dueling bandits were introduced by Yue et al. [2012]. The setting, pertaining to the K-armed bandit framework, assumes there is no direct access to the reward provided

by any single arm and the only information that can be gained is through the simultaneous pull of two arms: when such a pull is performed the agent gets access to the winner of the two arms, thus the name of *dueling* bandits. Here, we extend the framework of dueling bandits to the situation where there exist pairs of arms that are not comparable, that is, we study the case where there might be no natural order that could help decide the winner of a duel between two arms. A problem induced by such a framework is then to identify among the set of all available K arms the set of *maximal* incomparable arms, or the *Pareto front*, through a minimal number of pairwise pulls. To carry out our study, we propose to make use of tools from the theory of posets and we take inspiration from works dedicated to selection and sorting on posets Daskalakis et al. [2011].

Keys: Indistinguishability and Decoys. We make the assumption that the underlying poset or, more precisely, the incomparability structure, is not known. A pivotal issue that we have to face in this case is that of *indistinguishability*. In the bandit setting we assume, the draw of two arms that are comparable and that have close values —and hence a probability for either arm to win a duel close to 0.5— is essentially driven by the same random process, i.e. an unbiased coin flip, as the draw of two arms that are not comparable. Hence, if we denote by ε the distances between those two processes, we can have ε arbitrary small, and thus this pairs of arms cannot be distinguished from an incomparable pair of arm on the sole basis of pulls and a well-thought strategy. Such pair of arm will be referred as ε-indistinguishable. This problem has led us to make use of *decoy* arms. The idea of decoy originates from social psychology, and was originally intended to 'force' an agent (e.g., a customer) towards a specific good/action (e.g. a product) by presenting her a choice between the targetted good and a degraded version of it. Here, we use decoys to help solve the problem of indistinguishability

Contributions. Our main contribution, the UnchainedBandits algorithm, implements a strategy based on decoys and a peeling approach that finds the Pareto front[1] of a poset $\mathcal{S}$ with probability at least $1-\delta$ after at most $T \leq \mathcal{O}\left(K\frac{\mathbf{width}(\mathcal{S})}{\Delta^2}\log(NK^2/\delta)\right)$ pairwise pulls, while incurring a regret upper bounded by

$$\mathcal{R} \leq \frac{2K}{\gamma^2}\log\left(\frac{2NK^2}{\delta}\right)\sum_{i=1}^{K}\frac{1}{\Delta_i}C_{\alpha,\gamma}(N_i) + K\mathbf{width}(S)\log\left(\frac{2NK^2}{\delta}\right)\sum_{i,\Delta_i<\varepsilon_{N-1},i\notin\mathcal{P}}\frac{1}{\Delta_i},$$

where Δ is the parameter of the decoys, Δ_i the regret associated with arm i, K is the size of the poset, $\mathbf{width}(\mathcal{S})$ its *width*, N is the number of peeling iterations, γ is the peeling rate, ε_{N-1} is the maximum peeling and $C_{\alpha,\gamma}(N_i) \leq 1$ defined in Section 3 encodes the complexity of the poset with respect to arm i.

The paper is organized as follows. Section 2 presents the setting of dueling bandits on posets and formally states the problem we address. In Section 3, we formally introduce the notion of decoys and show how they can be constructed, both mathematically and practically, we then present our algorithm, UnchainedBandits, which relies on decoys, to find the *exact* Pareto front of the poset and we provide theoretical guarantees on their performances. In Section 4, we discuss how the present work relates to recent papers from the dueling bandits literature. Section 5 reports results on the empirical performances of our algorithm in different settings.

2 Problem Statement: Dueling Bandits on Posets

2.1 Reminders on Posets

We here recall base notions and properties about posets that are relevant to the present contribution.

Definition 2.1 (Poset)**.** *Let $\mathcal{S}$ be a set of elements. $(\mathcal{S}, \succcurlyeq)$ is a* partially ordered *or* poset *if $\succcurlyeq$ is a reflexive, antisymmetric and transitive binary relation on $\mathcal{S}$: $\forall a, b, c \in \mathcal{S}$*

- *$a \succcurlyeq a$ (reflexivity);*
- *if $a \succcurlyeq b$ and $b \succcurlyeq a$ then $a = b$ (antisymmetry);*

[1] We discuss in the supplementary material the easier setting where the incomparability information is known and we provide a dedicated algorithm, SlicingBandits, that takes advantage of this addiditional information.

- *if $a \succcurlyeq b$ and $b \succcurlyeq c$ then $a \succcurlyeq c$ (transitivity).*

Remark 2.2. *In the following, we will use $\mathcal{S}$ to denote indifferently the set $\mathcal{S}$ or the poset $(\mathcal{S}, \succcurlyeq)$, the distinction being clear from the context. We make use of the additional notation: $\forall a, b \in \mathcal{S}$*

- *$a \parallel b$ if a and b are* incomparable *(i.e. neither $a \succcurlyeq b$ nor $b \succcurlyeq a$);*
- *$a \succ b$ if $a \succcurlyeq b$ and $a \neq b$;*

Throughout, we limit our study to finite *posets, i.e., posets such that $|\mathcal{S}| < +\infty$.*

Definition 2.3 (Maximal element and Pareto front)**.** *An element $a \in \mathcal{S}$ is said to be a* maximal element *of $\mathcal{S}$ if $\forall b \in \mathcal{S}$, $a \succcurlyeq b$ or $a \parallel b$. We denote by*

$$\mathcal{P}(\mathcal{S}) \doteq \{a : a \succcurlyeq b \text{ or } a \parallel b, \forall b \in \mathcal{S}\},$$

the set of maximal elements or Pareto front of the poset.

Since there is no intrinsic reason to favor a particular maximal element, throughout this work we chose to focus on the task of finding the entire Pareto front $\mathcal{P}(\mathcal{S})$ or $\mathcal{P}$, for short. To this end, the notions of chain and antichain are key.

Definition 2.4 (Chain and antichain)**.** *$\mathcal{C} \subset \mathcal{S}$ is a* chain *(resp. an* antichain*) if $\forall a, b \in \mathcal{C},\ a \succcurlyeq b$ or $a \preccurlyeq b$ (resp. $a \parallel b$). $\mathcal{C}$ is said to be* maximal *if $\forall a \in \mathcal{S} \setminus \mathcal{C},\ \mathcal{C} \vee a$ is not a chain (resp. an antichain).*

Note that $\mathcal{P}$ is by definition a maximal antichain. Finally, the notion of *width* and *height* of a poset are important to characterize (the *complexity* of) a poset.

Definition 2.5 (Width and height)**.** *The width (resp. height) of a poset $\mathcal{S}$ is the size of its longest antichain (resp. chain).*

2.2 Dueling Bandits on Posets

K-armed Dueling Bandit. The K-armed dueling bandit problem [Yue et al., 2012] assumes the existence of K^2 parameters $\{\gamma_{ij}\}_{1 \leq i,j \leq K}$, with $\gamma_{ij} \in (-1/2, 1/2)$ and the following sampling procedure. At each time step, the agent pulls a pair of arms (i, j) and she gets in return the value of an independent realization of B_{ij}, a Bernouilli random variable with expectation $\mathbb{E}(B_{ij}) = 1/2 + \gamma_{ij}$ where $B_{ij} = 1$ means that i is the winner of the duel between i and j and, conversely, $B_{ij} = 0$ means that j is the winner. The objective of the agent is to find the *Condorcet winner* c—the arm such that $\gamma_{cj} > 0,\ \forall j \neq c$— among the K arms, whose existence is assumed, while minimizing the accumulated regret, defined for a sequence $((i_1, j_1), \dots, (i_T, j_T))$ of T pairs of pulls by $\frac{1}{2}\sum_{t=1}^{T}(\gamma_{ci_t} + \gamma_{cj_t})$.

Remark 2.6. *Note that: $\forall i, j,\ B_{ji} = 1 - B_{ij}$ and, thus, $\gamma_{ji} = -\gamma_{ij}$ and $\gamma_{ii} = 0$.*

The implicit assumption of traditional dueling bandits is that the set $\mathcal{S} = \{1, \dots, K\}$ of arms is totally ordered: for any pair $i, j \in \mathcal{S}$ of arms, i and j must be comparable and γ_{ij} *unequivocally* says which of the two is better.

Issues induced by working on posets. Now consider a dueling bandit problem defined on a poset $\mathcal{S}$. Compared to the usual setting where a total order on the arms exists, there are two main differences which arise when $\mathcal{S}$ is a poset: first, the situation where the agent pulls a pair of arms that are not comparable has to be handled with care and, second, there might be multiple maximal elements.

Working on bandits with a poset $\mathcal{S} = \{1, \dots, K\}$ of arms might be formalized as follows. For all chains $\{i_1, \dots, i_m\}$ of m arms there exist a family $\{\gamma_{i_p i_q}\}_{1 \leq p,q \leq m}$ of parameters such that $\gamma_{ij} \in (-1/2, 1/2)$; the pull of a pair of arms (i_p, i_q) from the same chain provides the realization of a Bernoulli random variable $B_{i_p i_q}$ with expectation $\mathbb{E}(B_{i_p i_q}) = 1/2 + \gamma_{i_p i_q}$. Regarding the incomparability, i.e. the situation where the pair of arms (i_p, i_q) selected by the agent correspond to arms such that $i_p \parallel i_q$, then there are two frameworks we propose to consider: one the one hand, the *fully observable posets*, where the draw from an incomparable pair of arms provides the agent with the information regarding the comparability of the arms[2]. On the other hand, that of *partially*

[2]This easier setting is analysed in depth in the supplementary materials.

observable posets, where such a draw is modeled as the toss of an unbiased coin flip—as we shall discuss, this framework poses the problem of indistinguishability mentioned in the introduction.

Regret on posets. In order to extend the notion of regret associated to an arm i, Δ_i, in the poset setting, we use the notion of distance to the Pareto front, noted $d(i, \mathcal{P})$ defined as follows :

$$\Delta_i = d(i, \mathcal{P}) = \min\{\gamma_{ij}, \forall j \in \mathcal{P} \text{ such that } j \succcurlyeq i\}.$$

We then define the regret occurred by comparing two arms i and j by $d(i, \mathcal{P}) + d(j, \mathcal{P})$. It is important to remark that the regret of a comparison is zero if and only if the agent is comparing two element of the Pareto front.

Problem statement. Given the issues induced by working on a poset $\mathcal{S}$ of arms, we may state that the problem that we want to tackle is to identify the Pareto front $\mathcal{P}(\mathcal{S})$ of $\mathcal{S}$ as efficiently as possible. More precisely, we want to devise pulling strategies for both poset observability frameworks such that for any given $\delta \in (0, 1)$, we are ensured that the agent is capable, with probability $1 - \delta$ to identify $\mathcal{P}(\mathcal{S})$ with controlled number of pulls *and* regret.

Assumption 1 (Order Compatibility)**.**

$$\forall i, j \in \mathcal{S}, \quad (i \succ j) \textit{ if and only if } \gamma_{ij} > 0.$$

We will not require any further hypothesis on how the γ_{ij} relate to each other and, therefore, no assumption on *strong stochastic transitivity* [Yue et al., 2012] is required.

2.3 Poset Observability

We consider the following setting, where the uncomparability information is not accessible.

Partially observable posets. A K-armed Dueling bandits on a partially observable poset $\mathcal{S} = \{1, \dots, K\}$ is a dueling bandit problem such that if $i \parallel j$, then $\gamma_{ij} = 0$. This property is referred as *Partial Observability*.

This property reflects the fact that neither of the two incomparable arms has a distinct advantage over the other: when the agent asks to compare two intrinsically incomparable arms, the results will only depend upon circumstances independent from the arms (like luck or personal tastes). Our encoding of such framework makes us assume that when considered over many pulls, the effects of those circumstances cancel out, so that no specific arm is favored, whence $\gamma_{ij} = 0$.

Consequences of partial observability. Note that partial observability entails the problem of indistinguishability evoked previously. Indeed, given two arms i and j, regardless of the number of comparisons, an agent may never be sure if either the two arms are very close to each other ($\gamma_{ij} \approx 0$ and i and j are comparable) or if they are not comparable ($\gamma_{ij} = 0$). Since all the elements of the Pareto set must be incomparable with each other, this renders the problem of identifying $\mathcal{P}$ intractable as well if no additional information is provided.

This problem motivates the following definition, which quantifies the notion of indistinguishability :

Definition 2.7 ($\varepsilon-$indistinguishability)**.** *Let $a, b \in \mathcal{S}$ and $\varepsilon > 0$. a and b are said to be ε-indistinguishable, noted $a \parallel_\varepsilon b$, if $|\gamma_{ab}| \leq \varepsilon$.*

As the notation $\parallel_\varepsilon$ implies, the $\varepsilon-$indistinguishability of two arms can be seen as a weaker form of incomparability, and note that as ε-decreases, previously indistinguishable pairs of arms become distinguishable, and the only $0-$indistinguishable pair of arms are the incomparable pairs. The classical notions of a poset related to incomparability can easily be extended to fit the $\varepsilon-$indistinguishability :

Definition 2.8 (ε-antichain, ε-width and ε-approximation of $\mathcal{P}$)**.** *Let $\varepsilon > 0$. $\mathcal{C} \subset \mathcal{S}$ is called an ε-antichain if $\forall a \neq b \in \mathcal{C}$, we have $a \parallel_\varepsilon b$. Additionally, $\mathcal{P}' \subset \mathcal{S}$ is called an $\varepsilon-$approximation of $\mathcal{P}$ if $\mathcal{P} \subset \mathcal{P}'$ and $\mathcal{P}'$ is an ε-antichain. Finally we denote by **width**$_\varepsilon(\mathcal{S})$ the size of the largest $\varepsilon-$antichain of $\mathcal{S}$.*

Interestingly, to find a $\varepsilon-$approximation of $\mathcal{P}$, it is only needed to remove the elements of $\mathcal{S}$ which are $\varepsilon-$distinguishable from $\mathcal{P}$. Thus, while $\mathcal{P}$ cannot be recovered in the partially observable setting,

Algorithm 1 Direct comparison

Given $(\mathcal{S}, \succ)$ a poset, $\delta, \varepsilon > 0$, $a, b \in \mathcal{S}$
Initialisation Maintains p_{ab} the average number of victory of a over b and I_{ab} its $1-\delta$ confidence interval,
Direct comparison:
while $0.5 + \varepsilon \in I$ or $0.5 - \varepsilon \in I$ **do**
  Compare a and b, Update p_{ab} and I.
  If $0.5 \notin I_{ab}$ and $p_{ab} > 0.5$, **Return** $a \succ b$; **Else If** $0.5 \notin I_{ab}$ and $p_{ab} < 0.5$, **Return** $b \succ a$.
end while
Return $a \parallel_\varepsilon b$

a $\varepsilon-$approximation of $\mathcal{P}$ can be obtained. Consequently, if the agent knows the minimum distance of any arm to the Pareto set, defined as $d(\mathcal{P}) = \min\{\gamma_{ij}, \forall i \in \mathcal{P}, j \in \mathcal{S} \setminus \mathcal{P}, \text{ such that } i \succ j\}$, she can recover the Pareto front, since for any $\varepsilon < d(\mathcal{P})$, the unique $\varepsilon-$approximation of $\mathcal{P}$ is $\mathcal{P}$ itself.

This information is however unavailable in practice and we choose not to rely on external information to solve the problem at hand. In the case where an ε approximation of the Pareto front is not enough, and the *exact* Pareto front is required, we devise an alternative strategy which rests on the idea of *decoys*, already mentioned in the introduction and fully developed in Section 3.

3 Contributions

Here, we introduce our algorithm, UnchainedBandits, that solves the problem of dueling bandits on partially observable posets, and we provide theoretical performance guarantees.

3.1 Decoys and Posets

As said in Section 2, deciding if two arms are incomparable or very close is intractable in the partially observable poset, and so is that of finding the *exact* Pareto front.

Still, without any additional device, the agent is able to find if two arms a and b, are ε-indistinguishable. using the *direct comparison* process provided by Algorithm 1. Yet, as previously discussed, this only produces an ε-approximation of the Pareto front, of whom $\mathcal{P}$ is only guaranteed to be a *subset*. To evade this shortcoming, we introduce a new tool, *decoys*, inspired by works from social psychology [Huber et al., 1982]. We formally define decoys for posets, and we prove that it is a sufficient tool to solve the incomparability problem (Algorithm 2). We also present methods for building those decoys, both for the purely formal model of posets and for real-life problems.

Definition 3.1 (Δ-decoy)**.** *Let $a \in \mathcal{S}$. Then $b \in \mathcal{S}$ is said to be a Δ-decoy of a if :*

1. $a \succcurlyeq b$ *and* $\gamma_{a,b} \geq \Delta$
2. $\forall c \in \mathcal{S}, a \parallel c$ *implies* $b \parallel c$
3. $\forall c \in \mathcal{S}$ *such that* $c \succcurlyeq a$, $\gamma_{c,b} \geq \Delta$

Interestingly, when $\mathcal{S}$ satisfies the strong stochastic transitivity hypothesis, the third point of the previous definition in an immediate consequence of the first. The following proposition illustrates how decoys can be used to determine the incomparability of two arms.

Proposition 3.2 (Decoys and incomparability)**.** *Let a and $b \in \mathcal{S}$. Let a' (resp. b') be a Δ-decoy of a (resp. b). Then a and b are comparable if and only if* $\max(\gamma_{b,a'}, \gamma_{a,b'}) \geq \Delta$.

Proof. Let us assume than $a \succcurlyeq b$. The transitivity of $\succcurlyeq$ implies that $a \succcurlyeq b'$, and the third point of Definition 3.1 implies that $\gamma_{a,b'} \geq \Delta$. The rest follows from point 2 of Definition 3.1. □

Algorithm 2 is derived from this result. The next proposition, an immediate consequence of Proposition 3.2, gives a theoretical guarantee on its performances.

Algorithm 2 Decoy comparison

Given $(\mathcal{S}, \succ)$ a poset, $\delta, \varepsilon > 0$, $a, b \in \mathcal{S}$
Initialisation Create a', b' the respective ε- decoy of a, b. Maintains p_{ab} the average number of victory of a over b and I_{ab} its $1 - \delta/2$ confidence interval,
Decoy comparisons:
while $0.5 + \varepsilon \in I$ **do**
  Compare a and b', b and a', Update p, and I.
  If $0.5 \notin I_{ab'}$ and $p_{ab'} > 0.5$, **Return** $a \succ b$. **Else If** $0.5 \notin I_{ba'}$ and $p_{ba'} > 0.5$, **Return** $b \succ a$.
end while
Return $a \parallel b$

Proposition 3.3. *Algorithm 2 returns the correct incomparability result with probability at least* $1 - \delta$ *after at most n comparisons, where* $n = 4\log(4/\delta)/\Delta^2$.

Adding decoys to a poset. A poset $\mathcal{S}$ may not contain all the necessary decoys. To alleviate this, the following proposition states that it is always possible to add relevant decoys to a poset.

Proposition 3.4 (Extending a poset with a decoy.)**.** *Let* $(\mathcal{S}, \succcurlyeq, \gamma)$ *be a dueling bandit problem on a partially observable poset, and* $a \in \mathcal{S}$. *Define* $a', S', \succ', \gamma'$ *as follows:*

- $\mathcal{S}' = \mathcal{S} \cup \{a'\}$
- $\forall b, c \in \mathcal{S}$, $b \succcurlyeq c$ *i.f.f.* $b \succcurlyeq' c$*,, and* $\gamma'_{b,c} = \gamma_{b,c}$
- $\forall b \in \mathcal{S}$, *if* $b \succcurlyeq a$ *then* $b \succcurlyeq a'$ *and* $\gamma'_{b,a'} = \max(\gamma_{b,a}, \Delta)$. *Otherwise,* $b \parallel a'$.

Then $(S', \succcurlyeq')$ *is a poset and* $(\mathcal{S}', \succcurlyeq', \gamma')$ *defines a dueling bandit problem on a partially observable poset,* $\gamma'_{|S} = \gamma$, *and* a' *is a* Δ*-decoy of* a.

Proof. The result naturally follows from the definition of a poset and Definition 3.1. □

Decoys in real-life problems. The intended goal of a decoy a' of a is to have at hand an arm that is known to be lesser than a. Creating such a decoy in real-life can be done by using a degraded version of a: for the case of a movie, a decoy can be obtain by e.g. decreasing the resolution of a film. Note that while for large values of the Δ parameter of the decoys Algorithm 2 requires less comparisons (see Proposition 3.3), in real-life problems, the second point of Definition 3.1 tends to becomes false: the new option is actually so worse than the original that the decoy becomes comparable (and inferior) to *all* the other arms, including previously non comparable arms (example: the decoy of a film for a very large $\Delta > 0$ could be in very low resolution such as 32×24; this film cannot be actually seen and is clearly worse than all the others, regardless of the genre). In that case, the use of decoys of arbitrarily large Δ can lead to erroneous conclusions about the Pareto front and should be avoided.

3.2 UnchainedBandits

We now present our algorithm, UnchainedBandits, that uses decoys to efficiently find the Pareto front of $\mathcal{S}$. UnchainedBandits is inspired by the ideas developed by Daskalakis et al. [2011], who address the problem of *sorting* a poset in a noiseless environment.

By Proposition 3.3, Algorithm 2 can be used to establish the exact relation between two arms. But this process can be very costly, as the number of required comparison is proportional to $1/\Delta^2$, even for strongly suboptimal arms. To avoid this possibility, UnchainedBandits implements a peeling technique: given $N > 0$ and a decreasing sequence $(\varepsilon_i)_{i=1}^{N-1}$ it computes and refines an ε_i-approximation of the Pareto front $\widehat{\mathcal{P}}_i$, using a subroutine (Algorithm 4), which considers ε_i-indistinguishable arms as incomparable. Then, at the N-th epoch, it uses Algorithm 4 one final time where it uses Algorithm 2 with Δ-decoys for comparisons, and then returns the Pareto front.

Algorithm subroutine. Algorithm 4 called on $\widehat{\mathcal{S}}$ with parameter $\varepsilon > 0$, $\delta > 0$ and $\mathcal{A}$ works as follows. It chooses a single initial *pivot*—an arm to which other arms are compared—and successively

Algorithm 3 UnchainedBandits

Given $\mathcal{S} = \{s_1, \dots, s_K\}$ a poset, $\delta > 0$, $\Delta > 0$, $N > 0$, $(\varepsilon_i)_{i=1}^{N-1} \in \mathbb{R}_+^N$
Initialisation Set $\mathcal{S}_1 = \mathcal{S}$, $\varepsilon_N = \Delta$.
Peel $\widehat{\mathcal{P}}$ **for** $t = 1$ **to** $N-1$ **do** $\mathcal{S}_{t+1}$ = `UBS Routine` $(\mathcal{S}_t, \varepsilon_t, \delta/N, \mathcal{A} =$ Algorithm 3). **end for**
Use decoys $\widehat{\mathcal{P}}$ = `UBS Routine` $(\mathcal{S}_N, \Delta, \delta/N, \mathcal{A} =$ Algorithm 2).
RETURN $\widehat{\mathcal{P}}$

Algorithm 4 `UBS Routine`

Given $\mathcal{S}_t$ a poset, $\varepsilon_t > 0$ a precision criterion, δ' an error parameter, $\mathcal{A}$ a comparison algorithm
Initialisation Choose $p \in \mathcal{S}_t$ at random. Define $\widehat{\mathcal{P}} = \{p\}$ the set of pivots.
Construct $\widehat{\mathcal{P}}$
for $c \in \mathcal{S}_t \setminus \{p\}$ **do**
  for $c' \in \widehat{\mathcal{P}}$ **do**
    Compare c and c' using $\mathcal{A}(\delta = \delta'/|S_t|^2, \varepsilon = \varepsilon_t)$. **If** $c \succ c'$, **Then** remove c' from $\widehat{\mathcal{P}}$.
  end for
  If $\forall c' \in \widehat{\mathcal{P}}$, $c \parallel c'$, **Then** add c to $\widehat{\mathcal{P}}$
end for
Return $\hat{\mathcal{P}}$

examines all the elements of $\widehat{\mathcal{S}}$. Each of the examined element p is compared to all the pivots. Each pivot that is dominated by p is removed from the pivot set. Then if after being compared to all the pivots, p was dominated by none, it is added to the pivot set. At the end, the set of remaining pivot is returned. During the first $N-1$ epochs, the comparisons are done with Algorithm 1. In the last epoch, the agent uses Algorithm 2 to get exact information on the relations between the remaining arms.

Reuse of informations. To optimize the efficiency of the peeling process, UnchainedBandits reuses previous comparison results. At the beginning of each *direct* comparison process between arms a and b, the empirical estimate p_{ab} and its confidence interval I_{ab} are initialized using the results of the previous direct comparisons of a and b. However, no information can be reused in the last epoch for the remaining arms, as the indirect comparison algorithm does not compare a to b directly.

The following theorem gives a high probability bound on the performances of UnchainedBandits.

Theorem 1. *The* UnchainedBandits *algorithm applied on $\mathcal{S}$ with parameters δ, Δ, N and with a decreasing sequence $(\varepsilon_i)_{i=1}^{N-1}$ lower bounded by $\Delta\sqrt{\frac{K}{\textbf{\textit{width}}(\mathcal{S})}}$, returns the Pareto front $\mathcal{P}$ of $\mathcal{S}$ with probability at least $1-\delta$ after at most T comparisons, with*

$$T \leq \mathcal{O}\left(K\textbf{\textit{width}}(\mathcal{S})\log(NK^2/\delta)/\Delta^2\right) \tag{1}$$

This is a consequence of the following intermediate result, whose proof can be found in the supplementary materials.

Proposition 3.5. *Algorithm 4 called on $\mathcal{S}_t$ with parameter $\varepsilon_t > 0$, $\delta' > 0$ and $\mathcal{A}$ = Algorithm 2 returns the Pareto front of $\mathcal{S}_t$ with probability at least $1-\delta'$ after at most*

$$T \leq 4|\mathcal{S}_t|\textbf{\textit{width}}(\mathcal{S}_t)\log(4|\mathcal{S}_t|^2/\delta')/\Delta^2$$

comparisons. Alternatively, when Algorithm 4 uses $\mathcal{A}$ = Algorithm 1, it returns an ε_t-approximation of the Pareto front of $\mathcal{S}_t$ with probability at least $1-\delta'$ after at most

$$T \leq 2|\mathcal{S}_t|\textbf{\textit{width}}_{\varepsilon_t}(\mathcal{S}_t)\log(2|\mathcal{S}_t|^2/\delta')\left(\frac{1}{\varepsilon_t^2} - \mathbf{1}_{t>1}\frac{1}{\varepsilon_{t-1}^2}\right)$$

additional *comparisons, where* $\mathbf{1}$ *is the indicator function.*

Proof of Theorem 1. Note that $\forall \mathcal{S}' \subset \mathcal{S}$ such that $\mathcal{P} \subset \mathcal{S}'$, $\mathcal{P}(\mathcal{S}') = \mathcal{P}$. The result is obtained by summing the upper bound in Proposition 3.5 over the different epochs, rearranging the sum and using the fact that $|\mathcal{S}_t|\textbf{width}_{\varepsilon_t}(\mathcal{S}_t)\log(N|\mathcal{S}_t|^2/\delta)$ is decreasing in t while $1/\varepsilon_t^2$ is increasing in t. The detailed proof can be found in the supplementary materials. □

Peeling rate. Note that even if **width**$(\mathcal{S})$ is unknown, it suffices to choose

$$\varepsilon_{N-1} \geq \Delta\sqrt{K} \tag{2}$$

to satisfy the hypotheses of Theorem 1. Although the previous result is valid for any decreasing sequence of (ε_t) satisfying (2), we focus on geometrically decreasing sequences, i.e. $\exists\gamma > 0$ such that $\varepsilon_t = \gamma^t\varepsilon_0$. For the sake of simplicity, we set $\varepsilon_0 = 0.5$ but all the following results can easily be extended for any $\varepsilon_0 > 0$.

Before we present our regret upper bound, we need to introduce a few notations. In order to characterise the efficiency of the peeling approach associated to γ, we define $\alpha(\gamma, \varepsilon_0, N) \in]0, 1]$, or α for short, as follows :

$$\alpha = \inf\left\{a \in [0,1],\ \text{s.t. } \forall 1 \leq t \leq N, \forall S_t \subset \mathcal{S} \text{ an } \varepsilon_0\gamma^t\text{-approximation of } \mathcal{P}, |S_t| \leq a^t|\mathcal{S}|\right\}. \tag{3}$$

It is important to note that the previous inequality is always true for $a = 1$, so α is always defined. α characterise how efficient is the peeling for the chosen parameters by quantifying the reduction in size between the successive $\mathcal{S}_t$. We can now introduce the following Theorem which gives an upper bound on the regret incurred by `Unchained Bandit`.

Theorem 2. *Let $\mathcal{R}_0$ (resp. $\mathcal{R}_1$) be the regret generated by Algorithm 3 applied on $\mathcal{S}$ with parameters δ, Δ, N and with a decreasing sequence $(\varepsilon_i)_{i=1}^{N-1}$ such that $\varepsilon_{N-1} \leq \Delta\sqrt{K}$ during the peeling phase (resp. the decoy phase). Let α as defined by (3). Then $\mathcal{R} = \mathcal{R}_0 + \mathcal{R}_1$ and with probability at least $1 - \delta$,*

$$\mathcal{R}_0 \leq \frac{2K}{\gamma^2}\log(\frac{2NK^2}{\delta})\sum_{i=1}^{K}\frac{1}{\Delta_i}C_{\alpha,\gamma}(N_i) \tag{4}$$

$$\mathcal{R}_1 \leq K\textbf{\textit{width}}(S)\log(\frac{2NK^2}{\delta})\sum_{i,\Delta_i<\varepsilon_{N-1}, i\notin\mathcal{P}}\frac{1}{\Delta_i}, \tag{5}$$

where

$$N_i = \min\left(\lceil\frac{\log(\Delta_i)}{\log(\gamma)}\rceil, N-1\right) \quad \textit{and} \quad C_{\alpha,\gamma}(n) = \begin{cases} n\alpha^{n-1} \textit{ if } \alpha = \gamma, \\ \dfrac{\gamma^{2n}(1-\alpha) + \alpha^n(\gamma^2-1)}{\gamma^2 - \alpha} \textit{ otherwise.} \end{cases}$$

In the previous theorem, N_i represent the number of peeling step where the arm i is present, while $C_{\alpha,\gamma}(N_i)$ represent the cost of doing the peeling for arm i. It is worth noting that $C_{\alpha,\gamma}(n) \leq 1$ and is increasing in α. This reflect the fact that small α are representative of an efficient pruning (many arms removed at each step).

Opposite constraints on ε. Theorem 1 is an upper bound on the number of comparisons required to find the Pareto front. This bound is tight in the (worst-case) scenario where all the arms are Δ-indistinguishable, i.e. peeling cannot eliminate any arm. In that case, any comparison done during the peeling is actually wasted, and the lower bound on ε_t (2) allows to upper bound the number of comparisons made during the peeling step to recover a K**width**(S) dependency in the upper bound, instead of K^2. On the other hand, a significant amount of peeling is required to obtain a reasonable upper bound on the incurred regret: the number of comparisons using decoys is very high ($\approx 1/\Delta^2$) and is the same for every arm, regardless of its regret. So it is important that only near-optimal arms remain during the decoy step, hence the upper bound on ε_t. In order to satisfy both constraints, ε_N must be chosen in $\left[\sqrt{K/\textbf{width}(S)}\Delta, \sqrt{K}\Delta\right]$.

4 Related Works

There is an actual connection between our work and studies from social psychology. In particular, Tversky and Kahneman [1981] issued one of the reference papers on the choice problem—which pertains to comparisons, in our framework— for real-life problems; they introduced the idea that alternatives may influence the perceived value of items. This idea had been taken one step further by Huber et al. [1982], who introduced and formalized the idea of decoys. They specifically argued that introducing *dominated alternatives*, i.e. decoys, may increase the probability of

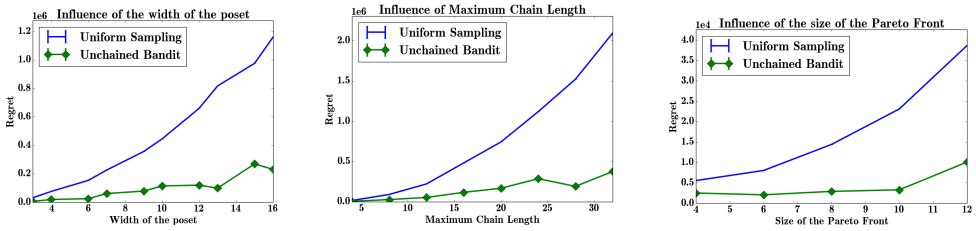

Figure 1: Time necessary to reach a conclusion for SlicingBandits and UnchainedBandits compared to UniformSampling, when the structure of the poset varies. Dependence on (left:) width of the poset, (center:) height of the poset and (right:) size of the Pareto front. Don't talk about SlicingBandits.

the original item to be selected: if A, B and A' are alternatives, then $\mathbb{P}(\text{select } A \text{ among } A, B) < \mathbb{P}(\text{select } A \text{ among } A, A', B)$. This generated an abundant literature (see Ariely and Wallsten [1995], Sedikides et al. [1999] and references therein) on works that studied the effect of decoys and their uses in various fields.

From the computer science literature, we must mention the work of Daskalakis et al. [2011], which addresses the problem on selection and sorting on posets and provides relevant data structures and accompanying analyses for computing on posets. Their results come down to classical results when totally ordered sets are used. Also, there might be yet other connections to draw between our work and that of Feige et al. [1994] who tackle the problem of sorting with noisy comparisons; note however that they assume there is a total order on the items they work on and the connection to be made with the present work would be to identify how this assumption may be weakened, if not removed.

Finally, we must discuss how our contribution separates from papers on dueling bandits. If the seminal paper of Yue et al. [2012] promotes algorithms, namely the Interleaved Filter algorithms, that exhibit optimal information-theoretic regret bounds, the authors assume the existence of a total order between the arms together as strong stochastic transitivity and (relaxed) stochastic triangle inequality. Since then, numerous methods have been proposed to relax those additional assumptions, including [Yue and Joachims, 2011, Ailon et al., 2014, Zoghi et al., 2014, 2015b]. Other approaches exist that do not assume the existence of a Condorcet winner, such as [Urvoy et al., 2013, Busa-Fekete et al., 2013, Zoghi et al., 2015a] but, to the best of our knowledge, we provide the first contribution that studies the framework where arms may be *incomparable*.

5 Numerical Simulations

In this section we experimentally evaluate UnchainedBandits. We did not compare our algorithm to dueling bandits algorithms from the literature, as a) they fail to consider the incomparability information and b) they are generally designed to return only one *best* element. Instead, we studied the performances of UnchainedBandits on simulated data (Section 5.1), and we applied it to an existing film rating database (Section 5.2).

5.1 Simulated Poset

First we confront UnchainedBandits with randomly generated posets, with different sizes, widths and heights. In order to give a baseline value, we use a simple algorithm, UniformSampling inspired from the successive elimination algorithm Even-Dar et al. [2006], which simultaneously compares all possible pairs of arms until one of the arms appears suboptimal, at which point it is removed from the set of selected arms. When only Δ-indistinguishable elements remain, it uses Δ-decoys.

Given the size $p > 0$ of the Pareto front, the desired width $w \geq p$ and the height $h > 0$, the posets are generated as follows: first, a Pareto front of size p is created. Then w chains of length $h - 1$ with no common elements are added. Finally, the top of the chains are connected to a random number of elements of the Pareto front. This creates the structure of the poset (i.e. the partial order $\succ$). Finally, the exact values of the γ_{ij}'s are obtained from a uniform distribution, conditioned to satisfy

Pareto Front	Highest average score
Pulp Fiction	Pulp Fiction
Fight Club	The Usual Suspect
The Shawshank Redemption	The Shawshank Redemption
The Godfather	The Godfather
Star Wars Episode V	The Godfather: Part II

Figure 2: Comparison between the five films with the highest average score (right column) and the five films of the $\varepsilon-$pareto set (left column)

the partially (or fully) observable framework. When needed, Δ-decoys are created according to Proposition 3.4. For each experiment reported on Figure 1, we changed the value of one parameter, and left the other to their default values ($p = 5$, $w = 2p$, $h = 10$). The results are averaged over ten runs. By default, we use $\delta = 1/1000$ and $\Delta = 1/100$. We also set $\gamma = 0.9$, $\varepsilon_0 = 0.5$ and $\varepsilon_N = \sqrt{K}\Delta$.

We note that for partially observable posets, UnchainedBandits produces much better results than UniformSampling and its advantage increases with the complexity of the problem.

5.2 MovieLens Dataset

To illustrate the example of the films recommendation system developed in the introduction, we chose to apply UnchainedBandits to the 20 millions items MovieLens dataset (Harper and Konstan [2015]).

To simulate a dueling bandit on a poset we proceed as follows: we remove all films with less than 50000 evaluations, thus obtaining 159 films, represented as arms. Then, when comparing two arms, we picke at random a user which has evaluated *both* films, and compare those evaluations (ties are broken with an unbiased coin toss). Since the decoy tool cannot be used in an already existing dataset, we restrict ourselves to finding an ε-approximation of the Pareto front, with $\varepsilon = 0.05$. Then, UnchainedBandits is run with parameters $\gamma = 0.9$, $\varepsilon_0 = 0.5$, $\varepsilon_N = \varepsilon$, $\delta = 0.001$.

There is no known ground truth for this experiments, so no regret estimation can be provided. Instead, the resulting Pareto front, which contains 5 films, is listed in Table 5.2, and compared to the five films among the original 159 with the highest average score. It is interesting to note that three films are present in both list, which reflects the fact that the *best* films in term of average score have a high chance of being in the Pareto Front. On the other hand, the films contained in the Pareto front are more diverse in term of genre, which is expected of a Pareto front. For instance, the sequel of the film "The Godfather" (hence very close to the original regarding genre) has been replaced by a a film of a totally different genre. It is important to remember that UnchainedBandits *does not have access to any information about the genre of a film* and its results are based solely on the pairwise evaluation of the user, and thus this result illustrates the effectiveness of our approach for the learning of the hidden poset.

6 Conclusion

We studied an extension of the dueling bandit problem to the poset framework, which raised the problem of ε-indistinguishability. We presented a new algorithm, UnchainedBandits, which tackles the partially observable settings, and we provided theoretical performance guarantee for its ability to identify the Pareto front. Future work might include the study of the influence of additional hypothesis on the structure of the poset, such as when the poset is actually a lattice or upper semi-lattice. In this case, different strategies of sampling might lead to even more efficient algorithms.

References

Nir Ailon, Zohar Karnin, and Thorsten Joachims. Reducing dueling bandits to cardinal bandits. In *Proceedings of The 31st International Conference on Machine Learning*, pages 856–864, 2014.

Dan Ariely and Thomas S Wallsten. Seeking subjective dominance in multidimensional space: An explanation of the asymmetric dominance effect. *Organizational Behavior and Human Decision Processes*, 63(3):223–232, 1995.

Róbert Busa-Fekete, Balazs Szorenyi, Weiwei Cheng, Paul Weng, and Eyke Hüllermeier. Top-k selection based on adaptive sampling of noisy preferences. In *Proceedings of The 30th International Conference on Machine Learning*, pages 1094–1102, 2013.

Constantinos Daskalakis, Richard M Karp, Elchanan Mossel, Samantha J Riesenfeld, and Elad Verbin. Sorting and selection in posets. *SIAM Journal on Computing*, 40(3):597–622, 2011.

Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *The Journal of Machine Learning Research*, 7:1079–1105, 2006.

Uriel Feige, Prabhakar Raghavan, David Peleg, and Eli Upfal. Computing with noisy information. *SIAM Journal on Computing*, 23(5):1001–1018, 1994.

F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(4):19, 2015.

Joel Huber, John W Payne, and Christopher Puto. Adding asymmetrically dominated alternatives: Violations of regularity and the similarity hypothesis. *Journal of consumer research*, pages 90–98, 1982.

Constantine Sedikides, Dan Ariely, and Nils Olsen. Contextual and procedural determinants of partner selection: Of asymmetric dominance and prominence. *Social Cognition*, 17(2):118–139, 1999.

Amos Tversky and Daniel Kahneman. The framing of decisions and the psychology of choice. *Science*, 211(4481):453–458, 1981.

Tanguy Urvoy, Fabrice Clerot, Raphael Féraud, and Sami Naamane. Generic exploration and k-armed voting bandits. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 91–99, 2013.

Yisong Yue and Thorsten Joachims. Beat the mean bandit. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 241–248, 2011.

Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.

Masrour Zoghi, Shimon Whiteson, Remi Munos, and Maarten D Rijke. Relative upper confidence bound for the k-armed dueling bandit problem. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 10–18, 2014.

Masrour Zoghi, Zohar S Karnin, Shimon Whiteson, and Maarten de Rijke. Copeland dueling bandits. In *Advances in Neural Information Processing Systems*, pages 307–315, 2015a.

Masrour Zoghi, Shimon Whiteson, and Maarten de Rijke. Mergerucb: A method for large-scale online ranker evaluation. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 17–26. ACM, 2015b.

Algorithm 5 SlicingBandits

Given $(\mathcal{S}, \succ)$ a poset with K elements, $\delta > 0$, $\mathcal{A}(.,.)$ a dueling algorithm with input a totally ordered set and a confidence value.
Initialisation Set $\widehat{\mathcal{S}} = \mathcal{S}$, $\widehat{\mathcal{P}} = \emptyset$.
while $\widehat{\mathcal{S}} \neq \emptyset$ **do**
  Extract a maximal chain from $\widehat{\mathcal{S}}$:
  Choose $p \in \widehat{\mathcal{S}}$ at random, initialize $\mathcal{C} = \{p\}$
  $\forall q \in \widehat{\mathcal{S}}$, if $\mathcal{C} \cup \{q\}$ is a chain, set $\mathcal{C} \leftarrow \mathcal{C} \cup \{q\}$
  $\widehat{\mathcal{S}} \leftarrow \widehat{\mathcal{S}} \setminus \mathcal{C}$
  Compute the maximal element of $\mathcal{C}$:
  Obtain $\hat{p} = \mathcal{A}(\mathcal{C}, \delta/K)$, update $\widehat{\mathcal{P}} \leftarrow \widehat{\mathcal{P}} \cup \{\hat{p}\}$
  Prune $\widehat{\mathcal{S}}$:
  $\forall q \in \widehat{\mathcal{S}}$, if $\hat{p}$ and q are comparable, update $\widehat{\mathcal{S}} \leftarrow \widehat{\mathcal{S}} \setminus \{q\}$
end while
RETURN $\widehat{\mathcal{P}}$

A Fully Observable Posets, SlicingBandits

Here we address the fully observable setting for the dueling bandits.

Fully observable posets. A K-armed Dueling bandit on a fully observable poset $\mathcal{S} = \{1, \dots, K\}$ is a dueling bandit problem such that if $i \parallel j$, and the agent pulls the pair (i, j), then the information of non-comparability is returned. This property is referred as *Full Observability*.

An efficient way to address this setting is to reconstruct the maximal chains of $\mathcal{S}$ by using the full observability property. Since every chain defines a total order, it is possible to use any total order Dueling Bandit algorithm on each of them. By carefully pruning the chain to avoid unnecessary comparisons, it is possible to efficiently recover the Pareto front of $\mathcal{S}$, with performances nearly as good as in the Totally order setting. This approach is detailed and analysed below.

Here, the agent may access the comparability information about any pair, and can thus retrieve the chains of $\mathcal{S}$. The following lemma states a simple property of maximal chains, that is essential to SlicingBandits.

Lemma A.1. *Every maximal chain $\mathcal{C}$ of a poset $\mathcal{S} \neq \emptyset$ contains a unique maximal element of $\mathcal{S}$.*

Proof. The result follows from the transitivity property of the poset. The complete proof can be found in the proof section of the supplementary material. □

Remark A.2. *By definition, it is easy to see that, conversely, for every maximal element p, there exists a (non-necessarily unique) maximal chain $\mathcal{C}$ of $\mathcal{S}$ such that $p \in \mathcal{C}$.*

To explore a chain in SlicingBandits the agent has to use a dueling bandit algorithm $\mathcal{A}$ devised for totally ordered set as a building block. We denote by $\mathcal{A}(\mathcal{C}, \delta)$ the maximal element of a totally ordered set $\mathcal{C}$ returned by $\mathcal{A}$ applied on the set $\mathcal{C}$ with confidence parameter δ.

Given $\mathcal{A}$, the agent proceeds as follows. She initializes $\widehat{\mathcal{S}} = \mathcal{S}$— $\widehat{\mathcal{S}}$ contains the elements that have not been processed yet—and $\widehat{\mathcal{P}} = \emptyset$, the candidates for the Pareto front, and, up until $\widehat{\mathcal{S}}$ is empty, the agent successively a) extract a maximal chain of $\widehat{\mathcal{S}}$, b) computes the maximal element of $\mathcal{C}$ (a totally ordered subset) by using $\mathcal{A}$, and c) prune $\widehat{\mathcal{S}}$, i.e. eliminates all the elements of $\widehat{\mathcal{S}}$ which are comparable to $\hat{p}$.

The upper bound on the number of pulls for SlicingBandits to provide the Pareto front is given by the next theorem.

Theorem 3. *Assume that $\mathcal{A}(\mathcal{C}, \delta')$ correctly returns the maximal element of $\mathcal{C}$ with probability at least $1 - \delta'$ using at most $\mathcal{T}(\mathcal{A}(\mathcal{C}, \delta'))$ pulls. Then* SlicingBandits *returns the Pareto front of $\mathcal{S}$ with*

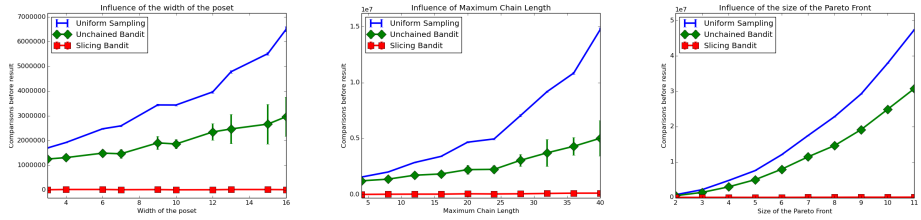

Figure 3: Time necessary to reach a conclusion for SlicingBandits and UnchainedBandits compared to UniformSampling, when the structure of the poset varies. Dependence on (left:) width of the poset, (center:) height of the poset and (right:) size of the Pareto front.

probability at least $1-\delta$ *with at most* $\mathcal{T}$ *comparisons, where*

$$\mathcal{T} \leq \mathcal{O}\left(K^2 + \sum_{c\in\mathcal{P}} \max_{\mathcal{C} \text{ a chain containing } c} \mathcal{T}(\mathcal{A}(\mathcal{C}, \delta\frac{|C|}{K}))\right).$$

Proof. The proof is divided into two parts: first, we only consider the event $\mathbf{E}_1$ where during the execution of Algorithm 5, each call to $\mathcal{A}(\mathcal{C}, \delta/\mathcal{S})$ returns the correct answer (the maximal element of $\mathcal{C}$), and we prove that on $\mathbf{E}_1$, Theorem 3 is correct. Second, using a bound on the number of calls to $\mathcal{A}$ performed on $\mathbf{E}_1$, we prove that $\mathbb{P}(\mathbf{E}_1) \geq 1-\delta$.

The following invariant holds on $\mathbf{E}_1$
Invariant: at the beginning of each iteration of the while loop, we have

$$\widehat{\mathcal{P}} \subset \mathcal{P} \subset \widehat{\mathcal{P}} \cup \widehat{\mathcal{S}} \text{ and } \forall p \in \widehat{\mathcal{P}}, \forall q \in \widehat{\mathcal{S}}, p \parallel q \tag{6}$$

A consequence is that $\widehat{\mathcal{P}}$ increases by one element at each iteration of the while loop, and thus $\mathcal{A}$ is called exactly $|\mathcal{P}|$ times, after which (6) implies $\widehat{\mathcal{P}} = \mathcal{P}$, hence

$$\mathbb{P}(\mathbf{E}_1^C) \leq |\mathcal{P}|\delta/|S| \leq \delta.$$

The number of additional comparisons required to build the chain is upper bounded by K^2, as all pairs of arms have to be compared at most once. Hence, the upper bound on $\mathcal{T}$ is derived from the fact that at each iteration, a chain with a different element of $c \in \mathcal{P}$ is considered. All the details of the proof can be found in the devoted section of the supplementary material. □

The following corollary illustrates Theorem 3 when $\mathcal{A}$ is the Interleaved Filter algorithm Yue et al. [2012].

Corollary A.1. *Assume that* $(\mathcal{S}, \succcurlyeq)$ *satisfies the strong stochastic transitivity and the triangle inequality of Yue et al. [2012]. Then* SlicingBandits *using the IF2 algorithm as* $\mathcal{A}$ *will return the correct Pareto front* $\mathcal{P}$ *with probability at least* $1-\delta$ *in at most* T *steps, where*

$$T \leq \mathcal{O}\left(K^2 + \frac{K}{(d(\mathcal{P}))^2}\log(K^2/\delta)\right).$$

Interestingly, when $\mathcal{S}$ is totally ordered, there is *one* maximal chain, $\mathcal{S}$, and SlicingBandits reduces to $\mathcal{A}$.

B Additional Numerical Simulations

The following experiments evaluate the relative efficiency of SlicingBandits and UnchainedBandits, we confront them with randomly generated posets, with different sizes, widths and heights.

Given the size $p > 0$ of the Pareto front, the desired width $w \geq p$ and the height $h > 0$, the posets are generated as follows: first, a Pareto front of size p is created. Then w chains of length $h-1$ with

no common elements are added. Finally, the top of the chains are connected to a random number of elements of the Pareto front. This creates the structure of the poset (i.e. the partial order $\succ$). Finally, the exact values of the γ_{ij}'s are obtained from a uniform distribution, conditioned to satisfy the partially (or fully) observable framework. When needed, Δ-decoys are created according to Proposition 3.4.

For each experiment reported on Figure B, we changed the value of one parameter, and left the other to their default values ($p = 5$, $w = 2p$, $h = 10$). The results are averaged over ten runs. By default, we use $\delta = 1/1000$ and $\Delta = 1/100$. The $(\varepsilon_t)_t$ are generated following the procedure presented in Section 3 with $\Delta_0 = 0.25$.

We did not compare our algorithms to dueling bandits algorithms from the literature, as a) they fail to consider the incomparability information and b) they are generally designed to return only one *best* element. Instead, we use a baseline algorithm, UniformSampling inspired from the successive elimination algorithm Even-Dar et al. [2006], which simultaneously compares all possible pairs of arms until one of the arms appears suboptimal, at which point it is removed from the set of selected arms. When only Δ-indistinguishable elements remain, it uses Δ-decoys.

We note that SlicingBandits clearly outperforms the other algorithms by a wide margin, thanks to the access to the comparability information and the careful management of chains. For partially observable posets, UnchainedBandits produces much better results than UniformSampling and its advantage increases with the complexity of the problem.

C Appendix : Extended Proofs

Proof of Lemma 3.1

Existence: Since $\mathcal{C}$ is a finite totally ordered set, it admits an unique maximal element. Let $c \in \mathcal{C}$ be the maximal element of $\mathcal{C}$. We use reductio ad absurdum. Suppose that c is not a maximal element of $\mathcal{S}$. By definition of maximal element, $\exists c' \in \mathcal{S}$ such that $c' \succ c$. But $\forall c'' \in \mathcal{C}$, we have $c \succcurlyeq c''$, then by transitivity $c' \succ c''$. Hence $\mathcal{C} \vee \{c'\}$ is a chain which strictly contains $\mathcal{C}$, which contradicts the fact the $\mathcal{C}$ is a maximal chain.

Uniqueness: let $c, c' \in \mathcal{C}$ be two maximal element of $\mathcal{S}$. Since $\mathcal{C}$ is a chain, c and c' are comparable. Since c is a maximal element, we have $c \succcurlyeq c'$. The same is true for c', hence the conclusion.

Proof of Theorem 1

Let $\mathbf{E}_1$ be the event where during the execution of Algorithm 1, each call to $\mathcal{A}(\mathcal{C}, \delta/K)$ return the correct answer (the maximal element of $\mathcal{C}$).

The proof is divided into two steps : First, we are going to prove that on $\mathbf{E}_1$, Theorem 1 is correct. Then, using an upper bound of the number of call to $\mathcal{A}$ done on the event $\mathbf{E}_1$, we will prove that $\mathbb{P}(\mathbf{E}_1^C) \leq \delta$, hence the conclusion.

On $\mathbf{E}_1$,consider the following invariant :
Invariant : At the beginning of each iteration of the while loop, we have

$$\widehat{\mathcal{P}} \subset \mathcal{P} \tag{7}$$

$$\forall p \in \widehat{\mathcal{P}}, \quad \forall q \in \widehat{\mathcal{S}}, \quad p \parallel q \tag{8}$$

$$\mathcal{P} \subset \widehat{\mathcal{P}} \cup \widehat{\mathcal{S}} \tag{9}$$

It is easy to see that the invariant is true at the beginning of the algorithm, because at the initialisation, $\widehat{\mathcal{P}} = \emptyset$ and $\widehat{\mathcal{S}} = \mathcal{S}$.

Assume that the invariant is true at the beginning of the loop $t + 1$, and denote by $\widehat{\mathcal{S}}_t$,$\widehat{\mathcal{P}}_t$ the value of $\widehat{\mathcal{S}}$, $\widehat{\mathcal{P}}$ at the end of loop t.

Since the algorithm has not stopped, $\widehat{\mathcal{S}}_t$ is not empty. By definition, the subset $\mathcal{C}$ constructed by the algorithm is a maximal chain of $\widehat{\mathcal{S}}$. Since $\mathcal{C}$ is a non empty finite totally ordered set, it admits an unique maximum element c.

We prove that

$$c \in \mathcal{P} \tag{10}$$

with reductio ad absurdum (RAA for short). Assume that $c \notin \mathcal{P}$. Then $\exists c' \in \mathcal{P}$ such that $c' \succ c$. Since $\mathcal{C}$ is a maximal chain of $\widehat{\mathcal{S}}$, it implies that $c' \notin \widehat{\mathcal{S}}$. Hence (9) implies that $c' \in \widehat{\mathcal{P}}$. But then $c' \succ c$ contradicts (8), which concludes the RAA.

Note that (10) and (8) implies

$$c \in \mathcal{P} \setminus \widehat{\mathcal{P}}. \tag{11}$$

Then, on $\mathbf{E}_1$, $\mathcal{A}(\mathcal{C}, \delta/K) = c$, and

$$\widehat{\mathcal{P}}_t \subsetneq \widehat{\mathcal{P}}_t \cup \{c\} = \widehat{\mathcal{P}}_{t+1} \subset \mathcal{P}. \tag{12}$$

Now by construction we have

$$\begin{aligned}\widehat{\mathcal{S}}_{t+1} &= \{p \in \widehat{\mathcal{S}}_t, \quad p \succ c \text{ or } p \parallel c\} \\ &= \{p \in \widehat{\mathcal{S}}_t, \quad p \parallel c\}\end{aligned}$$

since $c \in \mathcal{P}$. Then (8) implies that

$$\forall p \in \widehat{\mathcal{P}}_{t+1}, \quad \forall q \in \widehat{\mathcal{S}}_{t+1}, \quad p \parallel q. \tag{13}$$

Finally, we prove with RAA that

$$\mathcal{P} \subset \widehat{\mathcal{P}}_{t+1} \cup \widehat{\mathcal{S}}_{t+1} \tag{14}$$

Let $p \in \mathcal{P}$ such that $p \notin \widehat{\mathcal{P}}_{t+1} \cup \widehat{\mathcal{S}}_{t+1}$. (9) implies that $p \in \widehat{\mathcal{P}}_t \cup \widehat{\mathcal{S}}_t$. Since $\widehat{\mathcal{P}}_{t+1} \supset \widehat{\mathcal{P}}_t$, we have $p \in \widehat{\mathcal{S}}_{t+1} \setminus \widehat{\mathcal{S}}_t$. Then, by definition of $\widehat{\mathcal{S}}_{t+1}$, we have $c \succ p$, which contradicts $p \in \mathcal{P}$ and conclude the RAA.

Finally, (12)(13) and (14) implies that the invariant is true at the beginning of the loop $t + 2$.

When the algorithm stops, we have $\widehat{\mathcal{S}} = \emptyset$, hence (7) and (9) implies that

$$\widehat{\mathcal{P}} \subset \mathcal{P} \subset \widehat{\mathcal{P}} \cup \emptyset = \widehat{\mathcal{P}}$$

that is to say $\widehat{\mathcal{P}} = \mathcal{P}$. Hence on $\mathbf{E}_1$, Algorithm 5 reaches the correct conclusion.

A consequence of (11) is that $\widehat{\mathcal{P}}_t$ increases by exactly one element at each iteration of of the while loop, and thus the $\mathcal{A}$ is called exactly $|\mathcal{P}|$ times. Hence, if we denote by $\mathcal{C}_t$ the chain constructed at the loop t,

$$\begin{aligned}\mathbb{P}(\mathbf{E}_1^C) &\leq \sum_{t=1}^{|\mathcal{P}|} \mathbb{P}\left(\{\mathcal{A}(\mathcal{C}_t, \delta/K) \text{ failed }\}\right) \\ &\leq \sum_{c \in \mathcal{P}} \delta/K \leq \delta,\end{aligned}$$

Additionally, the number of additional comparisons required to build all the chains is upper bounded by K^2, as all pair of elements have to be compared at most once. Hence, the upper bound of $\mathcal{T}$ is derived from the fact due to (11), at each iteration, a chain with a different element of $c \in \mathcal{P}$ is considered.

Proof of Corollary 3.1

Let $\mathcal{C}_t$ be the chain considered by Algorithm 1 during the loop t, and we denote by c_t the maximal element of $\mathcal{C}_t$, which is the unique element of $\mathcal{P} \cap \mathcal{C}_t$ (consequence of (11)). Theorem 2 from

[Yue et al., 2012] implies that in this case,

$$\begin{aligned}\mathcal{T}\left(\mathcal{A}\left(\mathcal{C}_t, \delta/K\right)\right) &\leq \mathcal{O}\left(|C_t|\frac{\log(|C_t|^2K/\delta)}{(\min_{c'\in\mathcal{C}_t}\gamma_{c_tc'})^2}\right)\\ &\leq \mathcal{O}\left(|C_t|\frac{\log(K^3/\delta)}{(\min_{c\in\mathcal{P},c'\in\mathcal{S},c\succ c'}\gamma_{cc'})^2}\right).\end{aligned}$$

Using that by construction, $\forall t < t', \mathcal{C}_t \cap \mathcal{C}'_t = \emptyset$, and $\bigcup_t \mathcal{C}_t = \mathcal{S}$, we have

$$\begin{aligned}\sum_{t=1}^{|\mathcal{P}|}\mathcal{T}\left(\mathcal{A}\left(\mathcal{C}_t, \delta/K\right)\right) &\leq \mathcal{O}\left(\sum_{t=1}^{|\mathcal{P}|}|C_t|\frac{\log(K^3/\delta)}{(\min_{c\in\mathcal{P},c'\in\mathcal{S},c\succ c'}\gamma_{cc'})^2}\right)\\ &\leq \mathcal{O}\left(K\frac{\log(K^3/\delta)}{(\min_{c\in\mathcal{P},c'\in\mathcal{S},c\succ c'}\gamma_{cc'})^2}\right)\end{aligned}$$

Hence the conclusion.

Proof of Proposition 3.7

Case $\mathcal{A}$ = Algorithm 3
In this setting, the arms are compared using decoys.

We are going to proceed as in the proof of Theorem 1.

Let $\mathbf{E}_1$ be the event where during the execution of Algorithm 5, each call to Algorithm 3 returns the correct answer. We are going to prove the following invariant for the principal loop of the Algorithm on $\mathbf{E}_1$.

Invariant: At the iteration n, Let $\mathcal{S}_t^n$ the set of element of $\mathcal{S}_t$ already considered, $\widehat{\mathcal{P}}^n$ the current set of pivot. Then

$$\forall c' \in \mathcal{S}_t^n \quad \exists c \in \widehat{\mathcal{P}}^n, \quad c \succcurlyeq c' \tag{15}$$

$$\forall c, c' \in \widehat{\mathcal{P}}_t, \quad c \parallel c' \tag{16}$$

It is easy to see that the invariant is true at the beginning of the algorithm because $\mathcal{S}_t^0 = \widehat{\mathcal{P}}^0$ and $|\widehat{\mathcal{P}}^0| = 1$.

Suppose that the invariant is true at the n-th iteration. Let p be the new element considered, i.e. $\mathcal{S}_t^{n+1} = \mathcal{S}_t^n \cup \{p\}$, and define $\Gamma_-^p \doteq \{q \in \widehat{\mathcal{P}}^n, p \succ q\}$

1. **Case 1.** $\exists q \in \widehat{\mathcal{P}}^n$ s.t. $q \succ p$. In this case, $\widehat{\mathcal{P}}^{n+1} = \widehat{\mathcal{P}}^n \setminus \Gamma_-^p$, hence (16) at iteration n immediatly implies (16) at iteration $n+1$. Since $q \succ p$, we have $\forall q' \in \Gamma_-^p$, we have $q \succ q'$ by transitivity.Hence (15) at iteration n implies (15) at iteration n+1.
2. **Case 2.** $\forall q \in \widehat{\mathcal{P}}^n$, $p \succ q$ or $p \parallel q$. Then

   $$\widehat{\mathcal{P}}^{n+1} = \{p\} \cup \widehat{\mathcal{P}}^n \setminus \Gamma_-^p,$$

   and is it easy to see that (15) is still true iteration $n+1$. Now we are going to prove that (16) is still true by RAA. Assume that $\exists q \in \widehat{\mathcal{P}}^{n+1}$ s.t. q is comparable to p. By definition of Γ_-^p, it implies that $q \succ p$, which contradicts the initial assumption of the case.

After the last iteration n, we have $\mathcal{S}_t^{n+1} = \mathcal{S}_t$, since all the elements have been examined. We now prove by RAA that the invariant implies that $\widehat{\mathcal{P}}^{n+1} = \mathcal{P}$. We drop the $n+1$ in $\widehat{\mathcal{P}}^{n+1}$ for the sake of alleviating the notations.

Suppose that $\widehat{\mathcal{P}} \not\subset \mathcal{P}$ and let $p \in \widehat{\mathcal{P}} \setminus \mathcal{P}$. Since $p \notin \mathcal{P}, \exists q \in \mathcal{P}$ s.t. $q \succ p$. If $q \in \widehat{\mathcal{P}}$, (16) is contracted. Then $q \notin \widehat{\mathcal{P}}$. Hence $q \succ p$ contradicts (15). So $\widehat{\mathcal{P}} \subset \mathcal{P}$.

Now assume that $\mathcal{P} \not\subset \widehat{\mathcal{P}}$ and let $p \in \mathcal{P} \setminus \widehat{\mathcal{P}}$. Since $p \notin \widehat{\mathcal{P}}$, (15) implies that $\exists q \in \widehat{\mathcal{P}}$ s.t. $q \succcurlyeq p$. Since $p \notin \widehat{\mathcal{P}}$ and $q \in \widehat{\mathcal{P}}$, $q \neq p$ hence $q \succ p$, which contradicts $p \in \mathcal{P}$. So $\mathcal{P} \subset \widehat{\mathcal{P}}$. Hence $\widehat{\mathcal{P}} = \mathcal{P}$.

A consequence of (16) is that at each step, $\widehat{\mathcal{P}}^n$ is an antichain. Since during the execution of the algorithm all the elements of S_t are compared to all the element of the current $\widehat{\mathcal{P}}$, the algorithm do at most

$$|\mathcal{S}_t| \max_n |\widehat{\mathcal{P}}^n| \leq |\mathcal{S}_t| \text{width}(\mathcal{S}_t)$$

comparisons, and as a consequence

$$\mathbb{P}(\mathbf{E}_1^C) \leq |\mathcal{S}_t| \text{width}(\mathcal{S}_t) \frac{\delta}{|\mathcal{S}_t|^2} \leq \delta.$$

The upper bound on the number of comparisons results with the same remark combined with Proposition 3.5.

Case $\mathcal{A}$ = Algorithm 2.
During the epochs $t < N$, the arms are compared directly to each other, i.e. Algorithm 2 is used for comparisons purpose. We first tackle the case $t = 1$, i.e. the first epoch, since in this case, there is no previous observations, and thus no negative term in the upper bound.
Case $t = 1$. The proof for $t = 1$ unfolds similarly to the previous case, with a different invariant.

Let $\mathbf{E}_1$ be the event where during the execution of Algorithm 5, each call to Algorithm 2 returns the correct answer e.g. $i \succ j$ (resp $j \succ i$) if and only if $\gamma_{ij} > \varepsilon$ (resp $\gamma_{ji} > \varepsilon$). We are going to prove the following invariant for the principal loop of the Algorithm on $\mathbf{E}_1$.

Invariant: At the iteration n, Let $\mathcal{S}_t^n$ the subset of element of $\mathcal{S}_t$ already considered, $\widehat{\mathcal{P}}^n$ the current set of pivot. Then

$$\forall c' \in \mathcal{S}_t^n \quad \exists c \in \widehat{\mathcal{P}}^n, \quad c \succcurlyeq c' \tag{17}$$

$$\forall c, c' \in \widehat{\mathcal{P}}_t, \quad c \parallel_\varepsilon c' \tag{18}$$

It is easy to see that the invariant is true at the beginning of the algorithm because $\mathcal{S}_t^0 = \widehat{\mathcal{P}}^0$ and $|\widehat{\mathcal{P}}^0| = 1$.

Suppose that the invariant is true at the n-th iteration. Let p be the new element considered, i.e. $\mathcal{S}_t^{n+1} = \mathcal{S}_t^n \cup \{p\}$.

1. **Case 1.** $\exists q \in \widehat{\mathcal{P}}^n$ s.t. $q \succ p$ and $\gamma_{qp} > \varepsilon$. In this case, $\widehat{\mathcal{P}}^{n+1} = \widehat{\mathcal{P}}^n \setminus \Gamma_-^p$, hence (18) at iteration n immediatly implies (18) at iteration $n + 1$. Since $q \succ p$, we have $\forall q' \in \Gamma_-^p$, we have $q \succ q'$ by transitivity. Hence (17) at iteration n implies (17) at iteration $n + 1$.
2. **Case 2.** $\forall q \in \widehat{\mathcal{P}}^n$, ($p \succ q$ and $\gamma_{pq} > \varepsilon$) or $|\gamma_{pq}| < \varepsilon$. Then
$$\widehat{\mathcal{P}}^{n+1} = \{p\} \cup \widehat{\mathcal{P}}^n \setminus \Gamma_-^p,$$
and is it easy to see that (15) is still true iteration $n + 1$. Now we are going to prove that (16) is still true by RAA. Assume that $\exists q \in \widehat{\mathcal{P}}^{n+1}$ s.t. q is comparable to p and $|\gamma_{qp}| > \varepsilon$. By definition of Γ_-^p, it implies that $q \succ p$, and the order compatibility of the poset implies that $\gamma_{qp} > \varepsilon$ which contradicts the initial assumption of the case.

After the last iteration n, we have $\mathcal{S}_t^{n+1} = \mathcal{S}_t$, since all the elements have been examined. We now prove by RAA that the invariant implies that $\widehat{\mathcal{P}}^{n+1}$ is an ε-approximation of $\mathcal{P}$. We drop the $n + 1$ in $\widehat{\mathcal{P}}^{n+1}$ for the sake of alleviating the notations.

Now assume that $\mathcal{P} \not\subset \widehat{\mathcal{P}}$ and let $p \in \mathcal{P} \setminus \widehat{\mathcal{P}}$. Since $p \notin \widehat{\mathcal{P}}$, (15) implies that $\exists q \in \widehat{\mathcal{P}}$ s.t. $q \succcurlyeq p$. Since $p \notin \widehat{\mathcal{P}}$ and $q \in \widehat{\mathcal{P}}$, $q \neq p$ hence $q \succ p$, which contradicts $p \in \mathcal{P}$. So $\mathcal{P} \subset \widehat{\mathcal{P}}$.

Now suppose that $\exists q \in \widehat{\mathcal{P}}$ such that $\exists p \in \mathcal{P}$ s.t. $p \succ q$ and $\gamma_{pq} > \varepsilon$. Since $\mathcal{P} \subset \widehat{\mathcal{P}}$, we have $p \in \widehat{\mathcal{P}}$ and thus $\gamma_{pq} > \varepsilon$. contradicts (18). Hence $\widehat{\mathcal{P}}$ is a ε-approximation of $\mathcal{P}$.

A consequence of (18) is that at each step, $\widehat{\mathcal{P}}^n$ is an ε-antichain. Since during the execution of the algorithm all the elements of S_t are compared to all the element of the current $\widehat{\mathcal{P}}$, the algorithm do at most

$$|\mathcal{S}_t| \max_n |\widehat{\mathcal{P}}^n| \leq |\mathcal{S}_t| \mathbf{width}_\varepsilon(\mathcal{S}_t)$$

comparisons, and as a consequence

$$\mathbb{P}(\mathbf{E}_1^C) \leq |\mathcal{S}_t| \mathbf{width}_\varepsilon(\mathcal{S}_t) \frac{\delta}{|\mathcal{S}_t|^2} \leq \delta.$$

The upper bound on the number of comparisons results with the same remark combined with the fact that Algorithm 2 uses Hoeffding inequality.

Case $1 < t < N$**.**
To conclude, we only need to lower bound the number of previous comparisons that can be reused. Once again, consider the event $\mathbf{E}_1$ be the event where during the execution of Algorithm 5, each call to Algorithm 2 returns the correct answer e.g. $i \succ j$ (resp $j \succ i$) if and only if $\gamma_{ij} > \varepsilon$ (resp $\gamma_{ji} > \varepsilon$). Let i and $j \in \mathcal{S}_t$ such that i and j are compared at epoch t (i.e. during the call number t of Algorithm 3). Note that $S_t = \widehat{\mathcal{P}}_{t-1}^n$ and let assume without any loss of generality that i was added before j into $\widehat{\mathcal{P}}_{t-1}^n$. Since i is a pivot at the end of the epoch $t-1$, it was compared to all the arm considered after i, including j.

Since both i and j are pivots at the end of epoch $t-1$, it implies that $i \parallel j$ or $\gamma_{ij} < \varepsilon_{t-1}$. In both cases, Algorithm the algorithm does exactly $\frac{\log(K^2/\delta')}{\varepsilon_{t-1}^2}$ comparisons to reach this conclusion. The result follows from the reuse of information.

Proof of Theorem 2.

First note that if $\mathcal{P}'$ is a ε- approximation of $\mathcal{P}$, then $\mathcal{P} \subset \mathcal{P}'$. Additionally, it is easy to see that if $\mathcal{S}$ is a poset and $\mathcal{P}$ is its Pareto set, then $\forall \mathcal{S}' \subset \mathcal{S}$ such that $\mathcal{P} \subset \mathcal{S}'$, the Pareto front of $\mathcal{S}'$ is $\mathcal{P}$.

Hence, Proposition 3.7 implies that with probability at least $1 - N\delta/N = 1 - \delta$, Algorithm 4 returns the pareto front of $\mathcal{S}$. in at most T comparisons, where

$$\begin{aligned}
T &\leq 2 \sum_{t=1}^{N-1} |\mathcal{S}_t| \mathbf{width}_{\varepsilon_t}(\mathcal{S}_t) \log(2N|\mathcal{S}_t|^2/\delta) \left(\frac{1}{\varepsilon_t^2} - \mathbf{1}_{t>1} \frac{1}{\varepsilon_{t-1}^2} \right) + 4|\mathcal{S}_N| \mathbf{width}(\mathcal{S}_N) \frac{\log(4N|\mathcal{S}_N|^2/\delta)}{\Delta^2} \\
&\leq 2 \sum_{t=1}^{N-2} \frac{1}{\varepsilon_t^2} \left(|\mathcal{S}_t| \mathbf{width}_{\varepsilon_t}(\mathcal{S}_t) \log(2N|\mathcal{S}_t|^2/\delta) - |\mathcal{S}_{t+1}| \mathbf{width}_{\varepsilon_{t+1}}(\mathcal{S}_{t+1}) \log(2N|\mathcal{S}_{t+1}|^2/\delta) \right) \\
&\quad + \frac{2}{\varepsilon_{N-1}^2} |\mathcal{S}_{N-1}| \mathbf{width}_{\varepsilon_{N-1}}(\mathcal{S}_{N-1}) \log(2N|\mathcal{S}_{N-1}|^2/\delta) + 4|\mathcal{S}_N| \mathbf{width}(\mathcal{S}_N) \frac{\log(4N|\mathcal{S}_N|^2/\delta)}{\Delta^2}
\end{aligned}$$

where the second inequality is obtained by rearranging the sum. Now, by hypothesis we have

$$\varepsilon_t > \varepsilon_{N-1} \geq \Delta \sqrt{\frac{|\mathcal{S}|}{\mathbf{width}(\mathcal{S})}}$$

Hence, since the $|\mathcal{S}_t| \mathbf{width}_{\varepsilon_t}(\mathcal{S}_t) \log(N|\mathcal{S}_t|^2/\delta)$ is decreasing in t we have

$$\begin{aligned}
T &\leq 2 \sum_{t=1}^{N-2} \frac{\mathbf{width}(\mathcal{S})}{|\mathcal{S}|\Delta^2} \left(|\mathcal{S}_t| \mathbf{width}_{\varepsilon_t}(\mathcal{S}_t) \log(2N|\mathcal{S}_t|^2/\delta) - |\mathcal{S}_{t+1}| \mathbf{width}_{\varepsilon_{t+1}}(\mathcal{S}_{t+1}) \log(2N|\mathcal{S}_{t+1}|^2/\delta) \right) \\
&\quad + \frac{2\mathbf{width}(\mathcal{S})}{|\mathcal{S}|\Delta^2} |\mathcal{S}_{N-1}| \mathbf{width}_{\varepsilon_{N-1}}(\mathcal{S}_{N-1}) \log(2N|\mathcal{S}_{N-1}|^2/\delta) + 4|\mathcal{S}_N| \mathbf{width}(\mathcal{S}_N) \frac{\log(N|\mathcal{S}_N|^2/\delta)}{\Delta^2} \\
&\leq \frac{2}{\Delta^2} |\mathcal{S}| \frac{\mathbf{width}_{\varepsilon_1}(\mathcal{S})}{|\mathcal{S}|} \mathbf{width}(S) \log(2N|\mathcal{S}|^2/\delta) + 4|\mathcal{S}_N| \mathbf{width}(\mathcal{S}_N) \frac{\log(4N|\mathcal{S}_N|^2/\delta)}{\Delta^2} \\
&\leq \mathcal{O}\left(K \mathbf{width}(\mathcal{S}) \frac{\log(NK^2/\delta)}{\Delta^2} \right)
\end{aligned}$$

Proof of Theorem 3.

We know from Proposition 3.5, with probability at least $1-\delta$, the algorithm does not reach an incorrect result the comparison. For the rest of the proof, we restrict ourselves to this event.

First we consider the regret $\mathcal{R}_0$ induced by the peeling process.

Let i be an arm, and N_i be the last peeling step before i_p is eliminated. If i is not eliminated at the end of the peeling, then we set $N_i = N-1$. In other words,

$$\begin{aligned} N_i &= \max\{1 \leq t \leq N-1, \quad i \in \widehat{\mathcal{P}}_t\} \\ &= \min\left(\lceil \frac{\log(\Delta_i)}{\log(\gamma)} \rceil, N-1\right). \end{aligned}$$

Let $j \leq N_i$. During the j-th phase of peeling, the arm i is compared to at most $|S_j - 1|$ other arms. Hence, with the same argument as in Proposition 3.5, we have

$$\mathcal{R}_0 \leq 2\sum_{i=1}^{K} \Delta_i \sum_{t=1}^{N_i} |\mathcal{S}_t| \log(2N|\mathcal{S}_t|^2/\delta) \left(\frac{1}{\varepsilon_t^2} - \mathbf{1}_{t>1}\frac{1}{\varepsilon_{t-1}^2}\right).$$

Now since $\varepsilon_t < \varepsilon_{t-1}$, and by hypothesis, $|S_t| \leq \alpha^{t-1}K$, we have

$$\begin{aligned} \mathcal{R}_0 &\leq 2K\log(\frac{2NK^2}{\delta}) \sum_{i=1}^{K} \Delta_i \sum_{t=1}^{N_i} \alpha^{t-1} \left(\frac{1}{\varepsilon_t^2} - \mathbf{1}_{t>1}\frac{1}{\varepsilon_{t-1}^2}\right) \\ &\leq 2K\log(\frac{2NK^2}{\delta}) \sum_{i=1}^{K} \Delta_i \left(\sum_{t=1}^{N_i-1} \frac{\alpha^{t-1}}{\varepsilon_t^2}(1-\alpha) + \frac{\alpha^{N_i-1}}{\varepsilon_{N_i}^2}\right). \end{aligned}$$

Since by construction, we have $\varepsilon_{t+1} = \gamma\varepsilon_t$, then

$$\begin{aligned} \mathcal{R}_0 &\leq 2K\log(\frac{2NK^2}{\delta}) \sum_{i=1}^{K} \frac{\Delta_i}{\varepsilon_{N_i}^2} \left(\sum_{t=1}^{N_i-1} \gamma^{2(N_i-t)}\alpha^{t-1}(1-\alpha) + \alpha^{N_i-1}\right) \\ &\leq 2K\log(\frac{2NK^2}{\delta}) \sum_{i=1}^{K} \frac{\Delta_i}{\varepsilon_{N_i}^2} \left(\gamma^{2(N_i-1)} \sum_{t=1}^{N_i-1} (\frac{\alpha}{\gamma^2})^{t-1}(1-\alpha) + \alpha^{N_i-1}\right) \\ &\leq \frac{2K}{\gamma^2}\log(\frac{2NK^2}{\delta}) \sum_{i=1}^{K} \frac{1}{\Delta_i} \left(\gamma^{2(N_i-1)} \sum_{t=1}^{N_i-1} (\frac{\alpha}{\gamma^2})^{t-1}(1-\alpha) + \alpha^{N_i-1}\right), \end{aligned}$$

since by definition of N_i, we have $\Delta_i < \varepsilon_{N_i-1} = \gamma\varepsilon_{N_i}$. Now we have to consider two cases.

Case $\gamma^2 \neq \alpha$:

$$\begin{aligned} \mathcal{R}_0 &\leq \frac{2K}{\gamma^2}\log(\frac{2NK^2}{\delta}) \sum_{i=1}^{K} \frac{1}{\Delta_i} \left(\gamma^{2(N_i-1)}(1-\alpha)\frac{1-(\alpha/\gamma^2)^{N_i-1}}{1-(\alpha/\gamma^2)} + \alpha^{N_i-1}\right) \\ &\leq \frac{2K}{\gamma^2}\log(\frac{2NK^2}{\delta}) \sum_{i=1}^{K} \frac{1}{\Delta_i} \frac{\gamma^{2N_i}(1-\alpha) + \alpha^{N_i}(\gamma^2-1)}{\gamma^2-\alpha} \\ &\leq \frac{2K}{\gamma^2}\log(\frac{2NK^2}{\delta}) \sum_{i=1}^{K} \frac{1}{\Delta_i} C_{\alpha,\gamma}(N_i), \end{aligned}$$

and

Case $\gamma^2 = \alpha$:

$$\begin{aligned}
\mathcal{R}_0 &\leq \frac{2K}{\gamma^2}\log(\frac{2NK^2}{\delta})\sum_{i=1}^{K}\frac{1}{\Delta_i}\left(\alpha^{N_i-1}(1-\alpha)(N_i-1)+\alpha^{N_i-1}\right)\\
&\leq \frac{2K}{\gamma^2}\log(\frac{2NK^2}{\delta})\sum_{i=1}^{K}\frac{1}{\Delta_i}\alpha^{N_i-1}N_i\\
&\leq \frac{2K}{\gamma^2}\log(\frac{2NK^2}{\delta})\sum_{i=1}^{K}\frac{1}{\Delta_i}C_{\alpha,\gamma}(N_i),
\end{aligned}$$

hence the conclusion.

Now let $\mathcal{R}_1$ be the regret generated by the decoy step. To reach this step, an arm i must be such that $\Delta_i < \varepsilon_{N-1}$. If $i \in \mathcal{P}$, then pulling the arm i produces no regret. Otherwise, it is easy to see that the arm is compared to at most $\textbf{width}(S)$ other arms before being eliminated.

$$\begin{aligned}
\mathcal{R}_1 &\leq \textbf{width}(S)\log(\frac{2NK^2}{\delta})\sum_{i,\Delta_i<\varepsilon_{N-1},i\notin\mathcal{P}}\frac{\Delta_i}{\Delta^2}\\
&\leq (\frac{\varepsilon_{N-1}}{\Delta})^2\textbf{width}(S)\log(\frac{2NK^2}{\delta})\sum_{i,\Delta_i<\varepsilon_{N-1},i\notin\mathcal{P}}\frac{1}{\Delta_i}\\
&\leq K\textbf{width}(S)\log(\frac{2NK^2}{\delta})\sum_{i,\Delta_i<\varepsilon_{N-1},i\notin\mathcal{P}}\frac{1}{\Delta_i}.
\end{aligned}$$